Talk2Me: A Framework for Device–to–Device Augmented Reality Social Network

Jiayu Shu*, Sokol Kosta†, Rui Zheng*, and Pan Hui*‡
*HKUST-DT System and Media Laboratory, Hong Kong University of Science and Technology, Hong Kong
†Department of Electronic Systems, Aalborg University Copenhagen, Denmark
‡Department of Computer Science, University of Helsinki, Finland
jshuaa@ust.hk sok@cmi.aau.dk rzhengac@ust.hk panhui@ust.hk

***Abstract*—The continuous proliferation of mobile and wearable smart devices, together with their increasing computational power and multitude of sensors, has given birth to innovative applications that enhance the world with virtual layers of processed information. In this paper, we present Talk2Me, an augmented reality social network framework that enables users to disseminate information in a distributed way and view others' information instantly. Talk2Me advertises users' *messages*, together with their *face–signature*, to every nearby device in a Device–to–Device fashion. When a user looks at nearby persons through her camera–enabled wearable devices (e.g., Google Glass), the framework automatically extracts the face–signature of the person of interest, compares it with the previously captured signatures, and presents the information shared by this person to the user. We design and implement Talk2Me to be lightweight, given that it runs on mobile devices with limited power. We analyze different content dissemination strategies to find the best protocol that yields reliable and fast information–spreading, while reducing the number of packets and containing the energy consumption on the devices. We design a novel face recognition algorithm for this specific scenario with a small number of face features and limited computing capability. Evaluation results of the prototype with real users and extensive simulations validate the performance and usability of our design, showing the potentials of the augmented reality social network framework in real–world scenarios.**

I. Introduction

The continuous proliferation of mobile and wearable devices, the multitude of sensors and gadgets integrated in these devices, and advances in wireless communication technologies, have contributed to a rapid revolution on how people nowadays perceive the world and maintain social relationships. People and businesses use technology and social networks to share their ideas about politics, sport, art, etc.; to stay in touch with other users; or to advertise their professional profiles, interests, and products *to the world*.

In many scenarios though, people may be more interested in sharing quick information to people that are *physically nearby*. For example, shops or restaurants advertise their products or sales using big advertisement posters in order to let people know about their offers and attract potential clients. Movie theaters advertise movies using smart posters that allow users to extract additional content, e.g. a website or the movie trailer, by simply pointing their camera-enabled smart devices to the poster. Similarly, people use different methods to share information with nearby people. For instance, they announce their music taste by wearing T-shirts with the logo of their preferred music bands. Using these methods, however, people can only share static messages, without having the opportunity to elaborate or later modify them as they would do in a conversation or when posting in a social network.

Imagine a scenario where a technology workshop is held at an exhibition center. Start-up teams, investors, journalists, experts, and people offering or looking for jobs are participating. In front of a booth, *Alice*, who runs a start-up company, is presenting their products to people around. However, *Alice* would like to convey a specific information to a certain audience in an efficient way. For example, she wants to introduce the current situation and future plans of the company to investors, but she wants to communicate the technological details to developers. On the other hand, *Bob* is a student looking for a job, but is overwhelmed by the surrounding conversations and presentations. In this scenario, an instant and efficient social network for people in the physical proximity is needed. Though various types of online social networks have been used, they only link people that already know each-other, which cannot meet the demand of exchanging information to enhance social interactions with new people in the vicinity. Therefore, we need a new kind of pervasive social network that meets the following requirements and functionalities. *First*, the network can be created temporarily, without the need of any infrastructure or history records. *Second*, people can easily modify or update the information they want to deliver. *Third*, people are able to view and match the information with the person who sends it effortlessly, such as in an augmented reality way.

In this paper, we present *Talk2Me*, a novel framework for *Device-to-Device Augmented Reality Social Network* (*D2D–ARSN*) on mobile devices, which allows people to share messages with nearby users in a distributed way, and view others' shared information instantly. Previous works have already highlighted the usefulness of such distributed social networks. For example, E-SmallTalker connects proximate users based on their shared personal information [1]. Advancing on previous ideas, *Talk2Me* provides timely and dynamic information sharing and viewing functionalities that facilitate social interactions, especially for initiating conversations and making friends with people in physical proximity. Everyone using the framework is able to ***shout*** her thoughts, offering

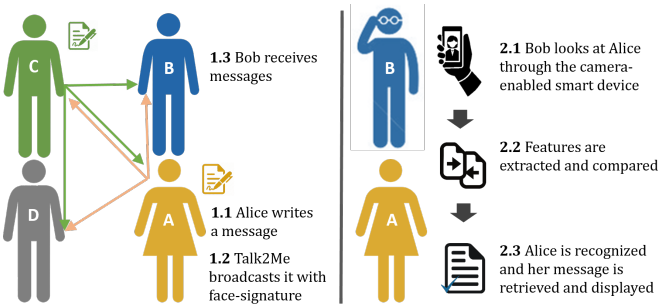


Fig. 1. A scenario that people are using Talk2Me.

small talk topics or useful messages to people in physical proximity for starting a conversation. Then, all people located nearby that have installed the framework will be able to ***listen to*** and ***view*** these messages through their camera–enabled devices, such as smart glasses or smartphones. For example, in the scenario shown in Figure 1, *Alice* broadcasts a message that she is looking for skilled mobile app developers and potential investors. *Bob* looks at people around through his camera–enabled device with Talk2Me installed, and sees *Alice's* message, so he goes to *Alice* for further information.

In the real world, we can identify a speaker by his voice or his face. Similarly, our framework uses visual features to uniquely differentiate message senders. When a message is sent, it also embeds unique face–signatures of the sender, extracted using state–of–the–art computer vision techniques [2], [3], [4], [5], so that the receiver can be able to associate the message with the person who originates it.

The main contribution of this paper is the proposal, design, and implementation of the first, to the best of our knowledge, distributed social network that works in augmented reality, by spreading information with embedded face–signatures. To this end, ***first***, we perform extensive simulations and laboratory tests to design a distributed protocol for message dissemination. The protocol has to be fast, to allow messages to be shared as quickly as possible among the users, and lightweight, considering that it will run on resource–limited mobile devices. ***Second***, we design and implement an accurate and lightweight face recognition algorithm which only requires a small number of face features per person, applying the state–of–the–art face feature extraction techniques. ***Third***, we perform quality of experience interviews with several test users to assess their level of satisfaction and to get feedback on how to further improve the framework.

The rest of the paper is organized as follows: in Section II, we introduce the related work; in Section III, we present a high–level overview of the framework; in Section IV, we describe its detailed design and implementation; in Section V, we present the evaluation results; in Section VI, we discuss concerns about security and privacy; and finally, in Section VII, we conclude the paper and discuss future work.

II. Related Work

Talk2Me adopts person identification techniques based on computer vision and relies on data exchange over D2D opportunistic networks, making it, to the best of our knowledge, the first known distributed augmented reality social network.

A. Vision Person Identification

Person identification via computer vision allows automatic recognition of an individual by using his visual traits. Face recognition, as the most popular identification technique, has been studied for years [2], [3]. Recently, Convolutional Neural Network (CNN) models have become the state–of–the–art on face recognition and feature extraction [4], [5], yielding the best results compared to other techniques. The models are usually pre–trained with hundreds of thousands of face images, therefore can be used for getting powerful face representations.

Though some other visual features can be obtained more easily, they are not as reliable as facial features in our scenario. For example, gait analysis focuses on walking properties such as silhouette sequences, step length, and speed, for individual recognition [6], [7]. Similarly, [8] and [9] present identification techniques that exploit people's clothing colors and temporary motion patterns like rotation, walking step duration, direction. However, these methods present problems in scenarios where people do not move. Therefore, they cannot be used to recognize individuals with reasonable accuracy.

B. D2D Peer Discovery and Communication

Device–to–Device (D2D) communication is a promising technology in the future social networks, which enables closely located devices to communicate with each other directly [10], [11]. To set up a D2D pair, peer discovery is the first step. In conventional D2D communication via cellular network, base stations help to identify D2D users and initiate D2D procedures. However, self-organized D2D networks are more flexible, since no infrastructure is needed [12]. Consequently, peer discovery is an important open issue in self-organized D2D networks, due to the lack of central controllers. One classic approach requires the user equipments to periodically broadcast HELLO messages and to respond immediately if they receive similar messages from the neighbors [13]. In this paper, we adopt the self-organized D2D networks for low–cost and convenient deployment, and design an efficient and lightweight protocol for D2D peer discovery.

Nowadays, D2D has been used in different scenarios and applications, such as: distributed content sharing, where co-located devices create, collect, and share content cooperatively [14], [15], [16]; D2D computation offloading, where devices help each other with remote procedure calls [17], [18]; message dissemination and content distribution in mobile social networks, where devices implement routing protocols to help sending messages from a *source* to a *destination* [19], [20], [21], [22]. Different from the above works, we apply D2D communication to augmented reality social networks by embedding face–signatures in the transmitted messages.

C. Mobile Social Networks in Proximity

Various approaches and architectures of mobile social networks in physical proximity have been surveyed in [23]. Initiating conversations and making friends with strangers in physical proximity is one of the most popular research ideas in this field. As of today, some prototypes have been implemented. For example, E-SmallTalker is a distributed mobile communications system that automatically discovers and suggests topics by performing information matching locally for people in physical proximity using Bluetooth [1]. E-Shadow is another distributed mobile phone-based local social networking system that allows users to perform layered information publishing and direction-driven localization to match the information with its owners [24]. Different from the existing mobile social networking systems, which focus mainly on profiles matching or localization, Talk2Me offers a more dynamic, convenient, and augmented reality way for sharing messages, finding the information of associated person, and displaying them in real time, thanks to the embedded face–signatures in the messages.

III. System Architecture

Talk2Me provides a seamless information–sharing channel that we believe will encourage social interactions in a crowd of people, improving their interaction efficiently without the need of any redundant registration process, creating an Augmented Reality Social Network (ARSN). We define two roles in the ARSN: *promulgator* and *recipient*. A promulgator is the person who broadcasts messages to nearby people. A recipient is the person receiving others' broadcasted information. Users can be both promulgator and recipient at the same time.

An overview of the system is presented in Figure 2. Its design and implementation is highly modular, consisting of five modules that we describe following a top–down approach.

Input Module: This module exposes different APIs for receiving messages from the user. If the message is a text for example, it opens a text field where the user can write the message, if it is an image it opens a picture browser, etc.

Camera Module: This module handles the interface used for *i)* getting visual data—i.e. capturing frames, and *ii)* for viewing the information associated with a face–signature. When a promulgator creates his face–signature, or a recipient wants to access the information shared by another person, the camera module is activated. This module captures the photo frames of the physical world and delivers them to the face module for further image processing. Upon receiving results from the face module, it displays them on the screen of a smartphone or smart glasses.

Face Module: This module is responsible for face detection, face–signature extraction, and face recognition. It works when a promulgator provides his/her face–signature for message preparation. Later, the promulgator can always use this pre–stored face–signature if he/she does not want to modify it. Moreover, this module is also involved when a recipient captures a frame. The module runs a face matching algorithm and retrieves the information associated with the matched

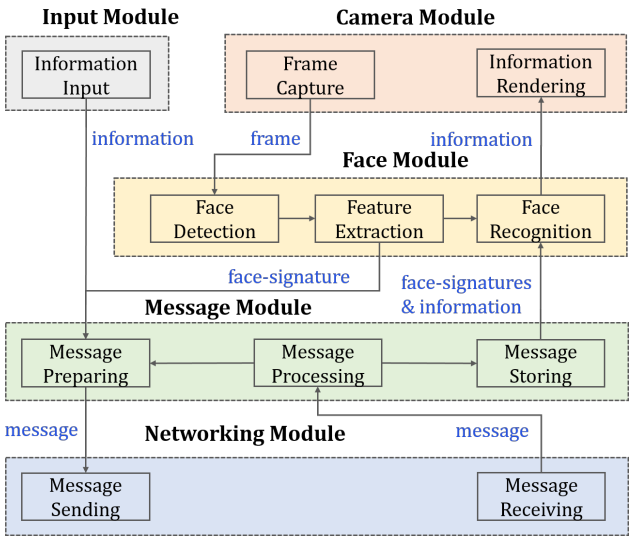


Fig. 2. Talk2Me architecture.

signature after face detection and feature extraction. Finally, the information is forwarded to the Camera Module.

Message Module: This module processes, prepares, and stores messages. On receiving a message from other devices through the networking module, it first processes the message in order to perform corresponding actions, such as storing the message internally or transmitting new messages. When necessary, this module gets the message from the input module and associates it with the face–signature of the promulgator, and then forwards the messages to the networking module.

Networking Module: This module receives messages from the message module and transmits them to the others. Meanwhile, it listens for incoming packets, which upon reception are passed to the message module.

IV. Design and Implementation

The most essential parts of the system are the *Image Processing Tasks*, which deal with face detection, face–signature extraction, face recognition, and the *Dissemination Protocol*, which deals with message transmission and reception.

In a scenario with N users, where all of them send messages to each other, there would be at least $N \times N$ messages needed for information spreading. Considering that users are not static, e.g. people coming late or leaving earlier, it is necessary for all devices to keep listening constantly for messages and to constantly advertise their presence. It is clear that sending messages very often would result in a quick device discovery, but it would come with a cost in terms of device overhead and energy consumption. These costs would not be reflected only on the transmitting devices but also on the receiving ones, since they would have to wake up and handle the messages.

The design of the image processing tasks is similarly crucial. The face detection, face–signature extraction, and face recognition should be fast, to allow for real–time user experience, and accurate, to correctly associate the face–signatures

with the messages. We implement and test our system on Android devices. In the next subsections we describe the details of the implementation deciding factors and the tools used to achieve the high–quality expected service.

A. Image Processing

The goal of image processing is to detect faces in the image frame, pre-process the face regions, extract face features, and prepare the face–signatures. In a promulgator's device, the processing module will save the face–signatures and use them for message transmission, while in a recipient's device the processing module will perform face recognition by comparing the face–signatures in the frame with those received.

Face Detection: It is the first step of image processing. We use open source Adaboost cascade classifiers [25] through the OpenCV library [26]. Moreover, we use skin color filters and the Dlib library [27] to reduce the number of false positives, further increasing face–detection precision.

Face Representation: It refers to the encoding of a face, which will be used as the *face–signature* of a user. It is important to notice that the encoding should be as small as possible in terms of bytes, since it will be used by promulgators for advertising. At the same time, it should accurately represent a face, since it will be used by recipients to associate faces with previously received face–signatures.

Traditionally, simple image descriptors such as *Local Binary Patterns* (*LBP*) [28] have been used extensively in the literature. Recently, feature extraction techniques based on *Convolutional Neural Networks* (*CNN*) have achieved state–of–the–art results in various computer vision tasks, including face recognition. We test both LBP and CNN models, and confirm that CNN models are more accurate. As a result, in this work we only present the results obtained using two CNN based models: *i)* a 4096–dimensional VGG–Face CNN descriptor, computed based on the VGG-Very-Deep-16 CNN architecture [4], pre-trained with $\sim 0.1M$ face images from 2622 subjects, and *ii)* a 256–dimensional features from a lightened–CNN model [5], pre-trained with $\sim 0.5M$ face images from 10575 subjects [29]. We use the last fully connected layer of each model, namely the $16KB$ output of the "*fc7*" layer from the VGG–Face CNN and the $1KB$ output of the "*eltwise_fc1*" layer from the lightened-CNN, as face representations. While CNN models with VGG architecture are more powerful and richer, in terms of tasks they can accomplish, than lightened–CNN models, their features are 16 times bigger. Indeed, as we show in the next sections, where we compare the two models, the smaller feature size makes the lightened–CNN model a good candidate for our system. To perform the feature extraction process we use the Caffe deep learning framework [30] and the Caffe Android library [31].

Face Recognition: It aims to associate a given face–signature with one that was received previously. As the system is totally decentralized, devices cannot rely on cloud services and have to perform the face recognition locally and independently on each device. Therefore, we identify three strategies on how this can be accomplished:

Algorithm 1 Face Recognition

initialize $Feature = f$, $Threshold = T$, number of people P, number of features per person Q, distance matrix D, candidate array C
for $i = 1$ to P **do**
  for $j = 1$ to Q **do**
    $D_{ij} \leftarrow d(f, f_{ij})$
    // $d(x, y)$ returns the distance between x and y
  end for
  $C_i \leftarrow \sum_j \theta(D_{ij} - T)$ // θ is unit step function
end for
Select top 2 elements of C, denote as C_r and C_s
if ($C_r > 0$ and $C_s = 0$) or ($C_r > 0$ and $C_s > 0$ and $C_r - C_s \geq Q/2$) **then**
  return r // person r is matched
else
  return Null // no face matched
end if

i) The first strategy would be to train the classifiers each time new face signatures are received. However, this solution presents a huge drawback, given that training the classifiers is a slow process and consumes a lot of energy.

ii) The second strategy would be to make each device broadcast *individual "pre-trained" classifiers*. When a receiving device performs face recognition on the extracted signature of the current photo, it runs all the received classifiers until finding the one that yields the best results, which would be the one trained by the respective promulgator with his photos. The drawback of this solution is that classifiers are usually much bigger than the face signatures in terms of bytes. Frequently sending large packets would hugely impact the system, especially in a scenario with multiple users.

iii) As a third strategy, we identify a solution where we do not need to train the classifiers when receiving new face signatures. In this case, we first calculate a similarity score between the extracted signature of the current image frame and the previously received face signatures from all the users to find the number of matches per person $C_i, i = 1, 2, \cdots, P$. Based on the number of matches we select two best candidates, whose number of matches are C_r and C_s. Only if the difference of the matches $|C_r - C_s|$ exceeds a certain margin, or if a single candidate has matches, we can conclude that the person is recognized. We set the margin as half the number of features per person after cross validation. As the previous two solutions are not practicable for our system, we implement the third strategy. The details of this face recognition procedure are shown in Algorithm 1.

B. Information Dissemination Protocol

We design and implement a distributed peer-to-peer protocol that allows all devices to deliver and receive information using D2D link within one-hop. We define three types of messages, depicted in Figure 3:

1) HELLO, used for advertising the presence of a user;
2) REQ, used to ask users to transmit their information; and
3) INFO, used to send the information.

Promulgators' devices can use HELLO messages to advertise their presence. This message consists of the promulgator's unique ID and his current information version. The ID is unique on each device and is generated from the SHA-256

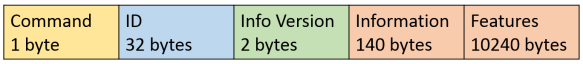


Fig. 3. Message format. *Command* is one of HELLO, REQ, INFO.

hash function of the device's IMEI. The information version is an integer that increases automatically whenever the user updates his message. Upon receiving the broadcasted HELLO message, a recipient's device will check if it already has this promulgator's information. If the recipient does not have the advertised information, it sends a REQ message to the promulgator. The REQ message contains the promulgator's ID and the promulgator's information version. The promulgator responds to a REQ message by sending an INFO message, which contains ID, information version, the actual information, and user's face–signature. In the current implementation, the information field is limited to 140 bytes, consisting of a message similar to a Tweet. The features field is limited to $10KB$, since it is the maximum size of the face–signatures we decide to use after performing extensive tests. It is clear that the INFO message is the heaviest of the three. This suggests that a good protocol should try to make use of the other types of messages as much as possible, so to contain the transmissions of the INFO messages at a minimum.

Dissemination Protocol: The objective of the information dissemination protocol is to make sure that devices get other users' INFO messages as fast as possible, while keeping the number of exchanged messages at a minimum. In this work, we compare three protocols. Protocol 1 is a simple broadcasting approach with no peer discovery (known as flooding or epidemic), while Protocol 2 is a typical peer discovery procedure in self-organized D2D networks. In order to reduce the number of HELLO messages sent by the promulgators, we propose Protocol 3, an improved version of the previous ones. Details of different protocols are described as follows:

Protocol 1. Promulgators broadcast INFO messages with a given frequency, e.g. every τ seconds. This will serve as a benchmark for the other more refined protocols, which try to reduce the number of messages, considering also the fact that INFO messages are very large compared to the other types.

Protocol 2. In this protocol we use HELLO and REQ messages so that devices can coordinate among each other using smaller packets before transmitting the large INFO messages. Devices broadcast HELLO messages with a fixed frequency, i.e. every τ seconds. Recipient devices that receive a HELLO and find they do not have the promulgator's current version of information, send a REQ message to the promulgator immediately. On receiving a REQ message, the promulgator responds by sending an INFO message to the asking device.

Protocol 3. Instead of broadcasting a HELLO message every τ seconds, the time interval between two messages follows a geometric progression with ratio α: $\tau, \tau\alpha, \tau\alpha^2, \ldots$, and so on. A device resets the interval to the initial value τ in three situations: *i)* if the user updates his face–signature or the information to advertise; *ii)* if a maximum threshold θ is reached, e.g. one hour; *iii)* or if it receives a HELLO_B message from an *unknown* device B. Specifically, when a device A receives a HELLO message from an unknown device B at time t, A will calculate the difference $\delta = t - t_i$, where t_i is the time that A broadcasted its last $HELLO_A$ message. If $\delta > \tau$, then A will broadcast a $HELLO_A$ message immediately.

We use the values $\tau = 60s$ in both protocols and $\theta = 1h$ in Protocol 3 in our evaluation experiments.

C. API to Applications and Developers

The Talk2Me framework serves as middle layer between client applications and the outside world. Applications that want to exchange messages between people in proximity by using users' face–signatures for identification can do so by exploiting the exposed API of Talk2Me. Moreover, given that the framework is highly modular, it can be easily extended to support multiple individual recognition techniques. Developers can include the new features into the framework and combine their results with the ones the framework already provides.

The following is a list of the most important methods implemented and exposed by the framework:

`setUserInfo():` setting or updating the information the user wants to share, such as text, URL link, etc.

`getFeatureFromCamera():` retrieving the extracted features from images or videos.

`broadcastMessage():` broadcasting the HELLO greeting messages.

`requestMessage():` sending the REQ message for requesting complete features and information.

`respondMessage():` sending the INFO message for responding to the received requests.

`getReceivedInfo():` retrieving a list containing the information transmitted by promulgators.

`matchFeature():` matching the received features with the given ones.

`renderInfo():` rendering information on the screen.

V. Evaluation

In this section, we present simulation and real testbed evaluations along three axes:

1) ***Image Processing***, where we evaluate the accuracy and efficiency of the VGG-Face CNN features and the lightened-CNN features on different distance metrics.

2) ***Information Dissemination Protocols Performance***, where we compare the number of messages and delay introduced in the system by each protocol, and the energy consumed by devices in different scenarios.

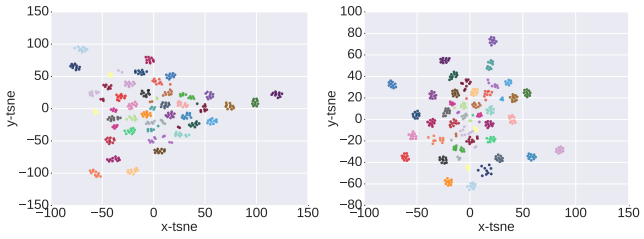
3) ***User Experience***, where we deploy the framework on a small–scale testbed with real users and interview them for feedback about their experience.

A. Experiment Setup

We implement a Java simulator to simulate the behavior of the information dissemination protocols in large–scale experiments. We also build a real prototype of the framework for Android devices to test the augmented reality features together

TABLE I
SPECIFICS OF THE TESTING DEVICES.

	Xiaomi Mi 3W	Galaxy Note 4	Xiaomi Mi 5
OS	Android 5.1.1	Android 6.0	Android 6.0
CPU	4×2.3 GHz	4×2.7 GHz	4×2.15 GHz
Chipset	Snapdragon 800	Snapdragon 805	Snapdragon 820
Camera	13 MP, f/2.2	16 MP, f/2.2	16 MP, f/2.0
RAM	2 GB	3 GB	4 GB



(a) VGG-Face CNN features (b) lightened-CNN features

Fig. 4. Two-dimensional visualization of two kinds of CNN face features.

with information dissemination in a real–world setting. We validate the implementation on three Android smartphones, shown in Table I. We implement the D2D communication over Wi-Fi Direct and over Wi-Fi. However, Wi-Fi Direct is still not quite mature and its D2D group formation technique burdens some of the devices more than the others [32]. To this end, we perform our experiment measurements in a D2D scenario where devices communicate with each other via UDP over Wi-Fi through an access point not connected to Internet.

B. Performance of Feature Extraction

Talk2Me heavily relies on the performance of face–signature extraction procedure. We evaluate the performance of this task on the three testing phones presented in Table I. We use the ORL Face Database [33], which consists of 400 images from 40 distinct subjects, 10 images per subject. Each subject has different photos, such as: with/without glasses, open/closed eyes, and different facial expressions. We extract face features from these images. Each feature is represented as an array of float values. In total, there are 10 feature arrays per subject, one from each different photo.

To assess the quality of feature extraction, we use t-SNE [34], a popular dimensionality reduction algorithm, to map each high dimensional feature array into a 2D point. The visual representation of the results are shown in Figure 4. We use 40 different colors to distinguish each subject. The results show that both lightened-CNN and VGG-Face CNN features have distinguishable structures. As we can see, even though VGG-Face CNN features tend to cluster more tightly, the lightened-CNN clusters are similarly distinguishable.

We compare the two CNN models on the three testing phones by measuring the time needed for feature extraction. We perform two different experiments: *i)* extracting 1 face feature array for each subject; and *ii)* extracting 10 face feature arrays per subject. In the second case, the 10 faces are processed in batch, at the cost of higher RAM usage. Each experiment is performed 10 times and the averaged results are summarized in Table II.

TABLE II
COST OF TIME FOR EXTRACTING FEATURES.

	Xiaomi Mi 3W	Galaxy Note 4	Xiaomi Mi 5
1 VGG-CNN	N/A	N/A	2780.3 ms
10 VGG-CNN	N/A	N/A	26740.0 ms
1 lightened-CNN	508.8 ms	330.0 ms	303.1 ms
10 lightened-CNN	6602.1 ms	3971.6 ms	2031.0 ms

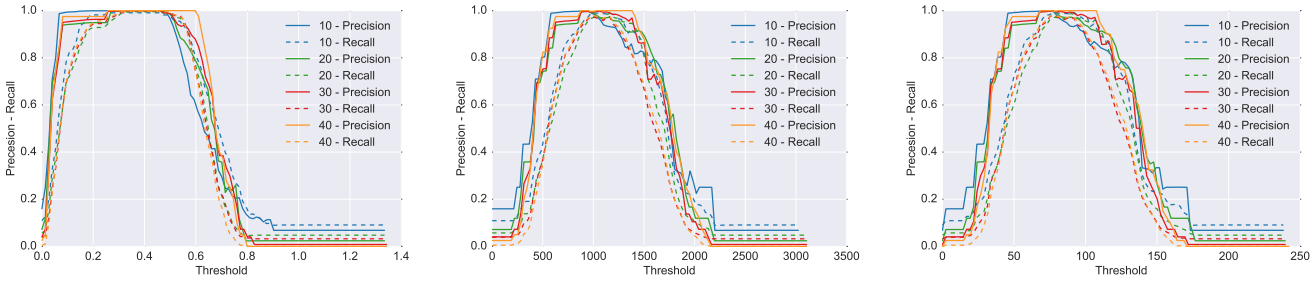
The lightened-CNN model is one order of magnitude faster than the VGG-Face CNN on all phones, independently by the number of features. The size of the lightened-CNN model is only $32.8MB$, much smaller than the VGG-Face CNN, which is $580MB$. From the table we can notice that we were unable to measure the time for the VGG-Face CNN on the Xiaomi Mi 3W and Galaxy Note 4 phones. This happens because the model consumes too much memory and the Android OS kills the application during the experiments. From these evaluations we conclude that the lightened-CNN model is the most suitable for our demands, satisfying all the needed requirements: accurate, lightweight, and fast. It is important to notice that the extraction of 10 features will be performed only when the user first opens Talk2Me or wants to update his face–signature. The extraction of one feature will be performed each time a user looks at a subject through the camera–enabled device for reading his messages. As we can see, the second operation takes less than half a second, which allows for near real–time user experience.

C. Accuracy of Face Recognition

As mentioned in Section IV, we do not use complex machine learning tools for face recognition, given that the task should be performed in real–time and is totally decentralized. Indeed, in this section we compare the accuracy of three distance functions, *Cosine*, *Manhattan*, and *Euclidean* on the features extracted on the previous dataset with 40 subjects.

We split the subjects into two groups: *database* and *query*, where the former represents the people whose features have been received and stored in a recipient's device, while the latter represents the people that the recipient is interested in reading their messages. The query group is composed of all 400 face images from 40 subjects, while the database group contains only a subset, representing the fact that the recipient's device has not received INFO messages from all users yet. Then, we compare the features extracted from each image in the query group with those in the database group. The goal of the distance functions is to match an image from the query group with one subject in the database group if they are the same.

We perform different experiments, varying the size of the database group $N \in \{10, 20, 30, 40\}$ subjects. Each of the N subjects represents a class c_i, where $i \in [1, 2, \ldots, N]$. We define a new class c_{N+1}, where we insert the subjects from the



(a) Cosine distance (b) Manhattan distance (c) Euclidean distance

Fig. 5. Precision–Recall using three different distance metrics on lightened-CNN features.

query group that do not belong to any class $c_i : 1 \leq i \leq N$. We denote as: *1) True Positives*: TP_i, the number of images from the query group that belong to class c_i and are correctly matched to class $c_i : 1 \leq i \leq N+1$; *2) False Negatives*: FN_i, the number of images from the query group that belong to class $c_i : 1 \leq i \leq N+1$, but are matched to a different class; and *3) False Positives*: FP_i, the number of images from the query group that do not belong to class c_i but are incorrectly matched to class $c_i : 1 \leq i \leq N+1$.

In Figure 5 we show the evaluation results of the accuracy of the distance metrics for different database group sizes. The plots show the results obtained with different threshold values for each function. To rate the output quality of the classifiers we use the *Precision-Recall* metrics, defined as:

$$\textbf{\textit{Precision}} = \left(\sum_{i=1}^{N+1} \frac{TP_i}{TP_i+FP_i}\right)/(N+1)$$

$$\textbf{\textit{Recall}} = \left(\sum_{i=1}^{N+1} \frac{TP_i}{TP_i+FN_i}\right)/(N+1),$$

where *Precision* is a measure of how many of the matched subjects are correct, while *Recall* is a measure of how many of the positive results were returned. The results clearly indicate that *Cosine* distance performs significantly better than the other two, as it shows an obvious plateau which has both high precision and high recall.

D. Performance and Overhead of Information Dissemination

In this section, we evaluate the performance and overhead of the information dissemination protocol via simulations.

Simulation Setup: We simulate two representative user–case scenarios: *i)* Events with fixed schedules, but no restrict attendance requirements, such as academic conferences, technical workshops, etc. Usually, the rate of arrival in these events initially is quite low, grows closer to the beginning, peaks right before it starts, and finally decays rapidly afterwards. We use normal distribution to model this situation. *ii)* Events without a specific starting time, such as public open days, exhibitions, etc. In these cases, we use Poisson distribution to model number of people arriving in a unit time.

In both cases we assume the arrival of 80 users in total, throughout 120 minutes of simulation. People who have already joined do not leave the event. We set the parameters of the normal distribution equal to $\mu = 60$ and $\sigma = 20$, meaning that most people arrive at $t \approx 60$. In Poisson distribution, we set 5 minutes as unit time and $\lambda = 3$, meaning 3 people every 5 minutes arrival rate.

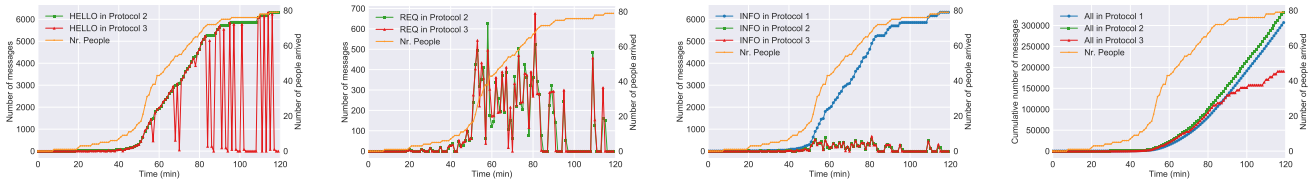
To evaluate the performance and overhead of the dissemination protocols, we measure the **number of messages** generated in the system during the whole time span, the **delay** in terms of time needed for a new user to obtain the INFO messages from each participating users, and the **energy** spent by the devices to communicate with each other.

Simulation Results: Figure 6 and Figure 7 show the number of HELLO, REQ, and INFO messages generated by Protocol 1, 2, and 3 in the simulated scenarios. The ratio parameter α in Protocol 3 is chosen to be 2. Figures *(a), (b)*, and *(c)* show the number of messages generated each minute, while figures *(d)* show the CDF of all the messages together.

As expected, Protocol 1 produces a large number of INFO messages and no HELLO and REQ messages (Figure 6(c) and 7(c)), while the number of INFO messages generated by Protocol 2 and 3 is highly contained.

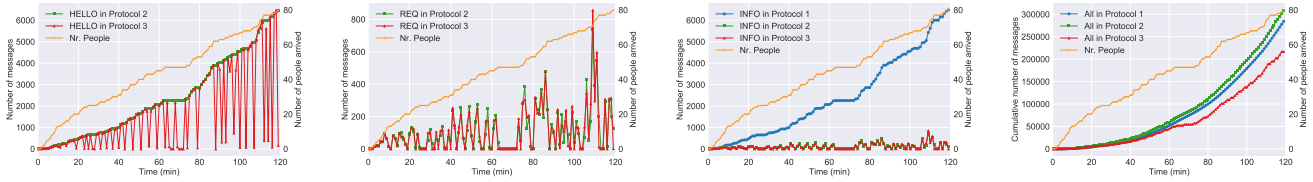
As for the HELLO messages, we can see how Protocol 3 clearly outperforms Protocol 2, reducing the number of these messages to zero when there are no new users joining. The two protocols behave almost the same when it comes to REQ and INFO messages. Protocol 3 replies to a HELLO message sent from a new user A immediately in most cases, by broadcasting a HELLO message, while Protocol 2 broadcasts the HELLO message with a regular frequency of 1 per minute, so it will wait until the 1–minute delay expires before sending its HELLO message. Nevertheless, both protocols will send a HELLO message within 1–minute interval, which will trigger the new user A to send a REQ message, which in turn will trigger the transmission of an INFO message.

From Figure 6(d) and 7(d), showing the CDF of all messages combined, we can see that Protocol 3 clearly generates less messages than the other two. Moreover, Protocol 3 is more sensible to the different scenarios, due to the different user arrival rates. While Scenario 2 presents a constant user arrival rate, Scenario 1 presents three different phases. Given that Scenario 1 presents more "quiet" phases, where people arrive more rarely, the number of messages is reduced sensibly in this scenario, which can be confirmed by looking at the CDF of the total number of messages in Figure 6(d) and 7(d).



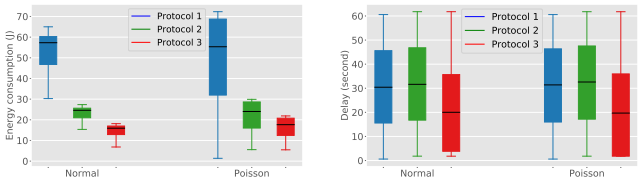
(a) HELLO messages (b) REQ messages (c) INFO messages (d) CDF of ALL messages

Fig. 6. Number of three types of messages in Scenario 1 following normal distribution of people arrival.



(a) HELLO messages (b) REQ messages (c) INFO messages (d) CDF of ALL messages

Fig. 7. Number of three types of messages in Scenario 2 following Poisson distribution of people arrival.



(a) Energy consumption (b) Delay

Fig. 8. Energy and INFO message delay.

TABLE III
ENERGY CONSUMPTION TO SEND AND RECEIVE DIFFERENT TYPES OF MESSAGES AS MEASURED ON THE SAMSUNG GALAXY NOTE 4 PHONE

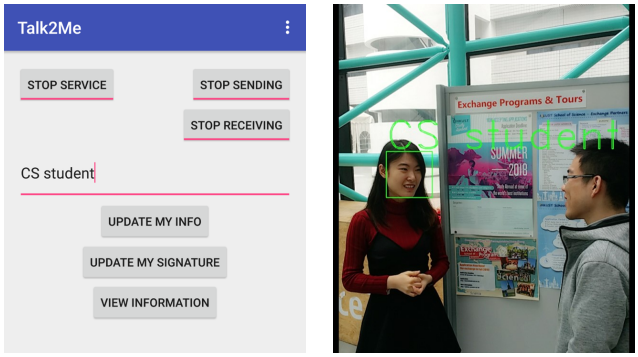
	HELLO		REQ		INFO	
	E (mJ)	std	E (mJ)	std.	E (mJ)	std
Sending	13.88	4.14	14.46	4.84	32.53	12.22
Receiving	4.44	2.33	4.34	2.09	13.12	6.60

In Figure 8(a) we show the energy consumption on all simulated devices during the entire simulation period. We present the results as a boxplot, showing the minimum, the 25^{th} percentile, the median, the 75^{th} percentile, and the maximum. To estimate these values we measure the energy consumed on the Galaxy Note 4 phone while sending and receiving different types of messages. Then, we multiply the measured values by the number of each message type sent or received by each device during the simulation. To measure the real energy, we perform a laboratory experiment by sending/receiving each message type 30 times, while monitoring the energy on the phone using the Monsoon Power Monitor [35]. Then, we calculate the average energy consumed over 30 measurements and the standard deviation, presented in Table III. Note that we do not measure the total energy consumption of the device, as we only want to compare the energy consumed by message transmission and receiving. As we can see from Figure 8(a), when using Protocol 3 devices consume four times and two times less energy than Protocol 1 and 2, respectively. Moreover, the total energy consumed in two hours simulation when using Protocol 3 is around $10-20J$, which corresponds to a very small fraction of the smartphone's battery capacity. Precisely, the Galaxy Note 4 has a 3000mAh, 3.85V battery, which contains $41.58KJ$ of energy if fully charged, meaning that less than 0.05% battery is consumed on each device by the information dissemination protocol in two hours. Notice that these measurements do not count the energy consumed by the camera or the image processing, which depend on the utilization of the framework.

Finally, Figure 8(b) shows the results of the delay, defined as "*time it takes for a new device joining the system at time i to receive the INFO message from a device that is already in the system*". We observe that Protocol 3 outperforms the other two even in this important metric, even though it is more conservative and produces less messages overall. This happens because devices following Protocol 3 reply to new HELLO messages within 1 minute, sometimes even immediately if their last HELLO message was broadcasted more than 1 minute ago, while in Protocol 1 and 2 they wait for the INFO and HELLO message timeout to expire, respectively.

E. User experience

We perform a real deployment in our campus where we recruit 25 mixed–gender participants (7 undergraduates, 1 postdoc, and 17 PhD students). We perform five experiments with groups of 2-5 users. First, we introduce the scenario in 2–5 minutes, showing how the system works with the interface as shown in Figure 9(a). In our prototype, we implement face recognition using Cosine distance with threshold $T = 0.4$, and Protocol 3 with $\alpha = 2, \tau = 60s, \theta = 1h$. We reset Talk2Me before giving the phones to the tester users, so that they could set their message and face signatures from scratch. An example of the result is presented in Figure 9(b). We let the testers use



(a) User interface (b) An exemplary screenshot

Fig. 9. Talk2Me prototype.

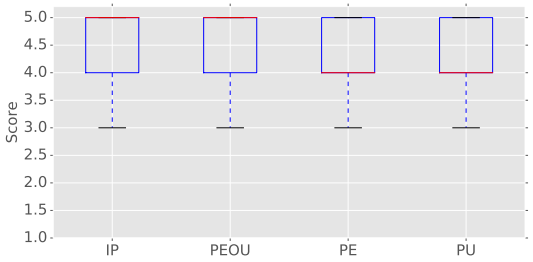


Fig. 10. User experience evaluation results.

the app for 5–10 minutes and then we ask them to answer four questions that we formulate according to a *Technology Acceptance Model (TAM)* [36], as follows:

1) Question about Image Processing (IP): *"How would you rate the performance of image processing, including face detection and recognition?"*.
2) Question about Perceived Ease Of Use (PEOU): *"How would you rate the ease of use about the prototype?"*.
3) Question about Perceived Enjoyment (PE): *"Overall, how would you rate your enjoyment when using it?"*.
4) Question about Perceived Usefulness (PU): *"How would you rate the usefulness of the prototype?"*.

The rating for each question ranges from 1 (very bad) to 5 (very good). As shown in Figure 10, the feedback for the 4 questions is quite positive. Most participants are positively surprised by the real–time augmented information viewing experience. Moreover, we ask them to provide suggestions or concerns about the prototype. Five raise privacy concerns related to the fact that personal face–signatures are transmitted to others. One participant considers the case of an attacker broadcasting bad messages or flooding the system with messages so that it could create a denial of service. Overall, we receive encouraging feedbacks, showing that the framework has large potential for deployment in large–scale scenarios.

VI. Discussion on Privacy Issues

There are several considerations on privacy issues of the Talk2Me framework that we need to discuss. *First*, the level of privacy mostly depends on users, as they broadcast messages they think are safe to be shared with nearby strangers. *Second*, the face-signature embedded in the message will not pose problems in terms of privacy, given that it is not possible to reconstruct the face image from such signatures.

However, there are also possible threats. An attacker could secretly relay and alter the messages from any promulgator, acting as a man–in–the–middle. Moreover, an attacker may create a flood of spam messages. But since the users only see the information when they look at the promulgators whose messages have been received, the spam messages will not be a problem for recipients. Another issue is that users of Talk2Me have to point their cameras towards other people faces, either using smartphones or smart glasses, in order to see their messages, which may make people feel uncomfortable. We are aware of these details and we are currently working on addressing them, to further improve Talk2Me.

VII. Conclusion and Future Work

In this work we designed, implemented, and evaluated Talk2Me, the first, to the best of our knowledge, distributed social network framework based on real–time augmented reality. Using Talk2Me, people can disseminate and receive information from people physically nearby in D2D fashion. The framework associates users' information with their individual face–signatures. When the camera of a device is pointed towards a person, the framework extracts the face–signature of the person and retrieves the messages he had advertised previously, showing them on the screen.

We designed an implemented a lightweight and yet highly accurate face recognition algorithm based on state–of–the–art CNN models. Moreover, we designed several distributed dissemination protocols that we evaluated by extensive simulations. We selected the best one, in terms of number of messages generated in the system, energy consumed by the devices, and delay of information spreading, and integrated its implementation in an Android prototype. We showed that using Talk2Me for two hours requires less than 1% of the total battery capacity to disseminate the information. To estimate the energy we measured the consumed energy of the single operations on real devices and fed the values to the simulator.

Finally, we performed preliminary real–world deployment with 25 subjects. Feedbacks from users were encouraging and helped us identify useful improvements that can be integrated in the framework. In particular, we will investigate on techniques that provide fine–grained data privacy and security. Moreover, we will perform simulations with more users and with more user arrival rate distributions. Finally, we will deploy the framework in large–scale real scenarios to assess its performance and scalability in the wild.

Acknowledgment

This research has been supported, in part, by projects 26211515 and 16214817 from the Research Grants Council of Hong Kong.

REFERENCES

[1] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*. IEEE, 2010, pp. 468–477.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[3] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in Face Detection and Facial Image Analysis*. Springer, 2016, pp. 189–248.

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, vol. 1, no. 3, 2015, p. 6.

[5] X. Wu, R. He, and Z. Sun, "A lightened cnn for deep face representation," *arXiv preprint arXiv:1511.02683*, 2015.

[6] J. Man and B. Bhanu, "Individual recognition using gait energy image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[7] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, "Gait recognition with kinect," in *1st international workshop on kinect in pervasive computing*. New Castle, UK, 2012, pp. P1–P4.

[8] H. Wang, X. Bao, R. R. Choudhury, and S. Nelakuditi, "Insight: recognizing humans without face recognition," in *Proc. 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 7.

[9] H. Wang, X. Bao, R. Roy Choudhury, and S. Nelakuditi, "Visually fingerprinting humans without face recognition," in *Proc. of 13th MobiSys*. ACM, 2015.

[10] S. A. Pambudi, W. Wang, and C. Wang, "On the resilience of d2d-based social networking service against random failures," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[11] X. Wang, H. Wang, K. Li, S. Yang, and T. Jiang, "Serendipity of sharing: Large-scale measurement and analytics for device-to-device (d2d) content sharing in mobile social networks," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2017, pp. 1–9.

[12] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, Fourthquarter 2014.

[13] S. Mumtaz and J. Rodriguez, *Smart device to smart device communication*. Springer, 2014.

[14] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, "Bluetorrent: Cooperative content sharing for bluetooth users," *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 609–634, 2007.

[15] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*. IEEE, 2011, pp. 136–146.

[16] S. Cho and C. Julien, "Chitchat: Navigating tradeoffs in device-to-device context sharing," in *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–10.

[17] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," in *Proc. of the 13th ACM MobiHoc*, 2012, pp. 145–154.

[18] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "Openrp: a reputation middleware for opportunistic crowd computing," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 115–121, 2016.

[19] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware for mobile social networking," in *Proc. of the 2nd ACM workshop on Online social networks*. ACM, 2009, pp. 49–54.

[20] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '08. New York, NY, USA: ACM, 2008, pp. 241–250. [Online]. Available: http://doi.acm.org/10.1145/1374618.1374652

[21] K. C.-J. Lin, C.-W. Chen, and C.-F. Chou, "Preference-aware content dissemination in opportunistic mobile social networks," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1960–1968.

[22] Y. Li and W. Wang, "Message dissemination in intermittently connected d2d communication networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3978–3990, 2014.

[23] Y. Wang, A. V. Vasilakos, Q. Jin, and J. Ma, "Survey on mobile social networking in proximity (msnp): approaches, challenges and architecture," *Wireless networks*, vol. 20, no. 6, pp. 1295–1311, 2014.

[24] J. Teng, B. Zhang, X. Li, X. Bai, and D. Xuan, "E-shadow: Lubricating social interaction using mobile phones," *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1422–1433, 2014.

[25] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[26] "OpenCV," http://opencv.org/, 2017.

[27] "Dlib," http://dlib.net/, 2017.

[28] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[29] "Casia-webface dataset," http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html, 2017.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[31] sh1r0, "Caffe-android-lib," https://github.com/sh1r0/caffe-android-lib, 2017.

[32] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, June 2013.

[33] "The orl face database," http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, 2017.

[34] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[35] "Monsoon power monitor," http://https://www.msoon.com/LabEquipment/PowerMonitor/, 2017.

[36] A.-C. Haugstvedt and J. Krogstie, "Mobile augmented reality for cultural heritage: A technology acceptance study," in *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, 2012.